A System for Indian Postal Automation

Kaushik Roy, Szilárd Vajda, Umapada Pal, Bidyut Baran Chaudhuri, Abdel Belaïd

A System for Indian Postal Automation

K. Roy[+], S. Vajda[*], U. Pal[+], B. B. Chaudhuri[+] and A. Belaid[*]
[+]*Computer Vision and Pattern Recognition Unit Indian Statistical Institute, Kolkata-108, India*
[*]*LORIA Research Center, B.P. 239, 54506, Nancy, France*

Abstract

In this paper, we present a system towards Indian postal automation based on the recognition of pin-code and city name of the postal document. In the proposed system, at first, non-text blocks (postal stamp, postal seal etc.) are detected and Destination Address Block (DAB) is identified from the document. Next, lines and words of the DAB are segmented. Since India is a multi-lingual and multi-script country, the address part may be written by combination of two scripts. To identify the script by which a word is written, we propose a water reservoir based technique. It is very difficult to identify the script by which the pin-code portion is written. So we have used two-stage artificial Neural Network (NN) based general classifiers for the recognition of pin-code digits written in English/Bangla. For recognition of city names we propose an NSHP-HMM (Non-Symmetric Half Plane-Hidden Markov Model) based technique.

1. Introduction

Postal automation is a topic of research interest for last two decades and many pieces of published article are available towards automation of non-Indian language documents [1-3]. Several systems are also available for address reading in USA, UK, France, Canada and Australia. But no system is available for address reading of Indian postal documents.

System development towards postal automation for a country like India is more difficult than that of other countries because of its multi-lingual and multi-script behavior. Some people write the destination address part of a postal document in two or more language scripts. For example, see Fig.1, where the destination address is written partly in Bangla script and partly in English. In India there is a wide variation in the types of postal documents. Post-card, inland letter, special envelopes are sold from Indian post offices and there is a pin-code box to write pin number and also some commercial envelopes with or without pin-code box. In some documents we may find partial pin code instead of full pin-code and even no pin-code. For example Kol-32 is written instead of Kolkata-700032. Thus, development of Indian postal address reading system is a challenging problem.

In this paper, we propose a system towards Indian postal automation where at first, using Run Length Smoothing Approach (RLSA) and characteristics of different image components, the postal stamp/seal parts are detected and removed from the documents. Next, based on positional information, the DAB region is located. After extraction of pin-code box from DAB region, pin-code numerals written within the pin-code box are extracted. Using a two-stage NN, the Bangla and English numerals of the pin-code part are recognized. We have seen that in 36.2% of the cases the documents pin-code is either absent or partially written. For such documents, we need to recognize the city name or post office name. For this purpose we first segment DAB into lines and words. Next using water reservoir concept based feature word-wise script identification is done. After a differential height normalization of word images, a context based, fully 2D NSHP-HMM approach has been used to recognize words.

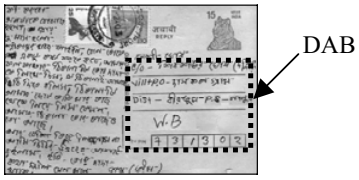

Fig.1. Example of bi-script postal document. DAB is shown by dotted square.

2. Preprocessing

2.1. Data collection and noise removal

Document digitization for the present work has been done from 7500 real life data collected from an Indian post-office. We have used a two-stage approach to convert them into two-tone (0 and 1) images [4]. The digitized document images may be skewed and we used Hough transform to de-skew the documents [4].

2.2. Postal stamp detection and DAB detection

The binary image is processed to extract the postal stamp and other graphics parts present in the image. Here, we used a combined technique as follows. We

Proceedings of the 2005 Eight International Conference on Document Analysis and Recognition (ICDAR'05)
1520-5263/05 $20.00 © 2005 IEEE

first smooth the image using RLSA [4]. On this smoothed image we apply component labeling to get individual blocks. For each smoothed block we find its boundary and check the density of black pixels and count the number of components over the corresponding boundary area on the original binary image. Based on the property that postal stamp/seal block has high density of black pixels and/or contains many small components compared to that of text portion, graphics parts are detected. Using positional information the DAB is extracted from the document.

2.3. Script identification

Because of multi-lingual and multi-script behavior a single line of a postal document may be written by more than one script. To correctly recognize a word it is necessary to feed it to the OCR of that script in which the word image is written. So, we have to identify the script of (Bangla or English script in present case) each word. Here we used piece-wise projection method [4] for text line and word segmentation. Following features are used for word-wise script identification.

2.3.1. Water reservoir principle based feature: The principle of water reservoir property is as follows. If water is poured from a side of a component, the cavity regions of the component where water will be stored are considered as reservoirs [4]. By top (bottom) reservoir of a component we mean the reservoirs obtained when water is poured from top (bottom) of the component. For examples see Fig.2(c).

While writing, characters in a word generally touch one another and create a large space and this space generates reservoir. Sometimes in a word all its characters does not touch each other. As a result, we may not get proper water reservoirs and our scheme may not work properly. So, we join the components of a multi component word shown in Fig.2(a-b). Details about joining see [4].

The features calculated based on water reservoir principle are ratio (r) of the area of the top to that of bottom reservoirs and the base line (the line passing through the average of all the base (the deepest point) points of the bottom reservoirs). For illustration see Fig.3.

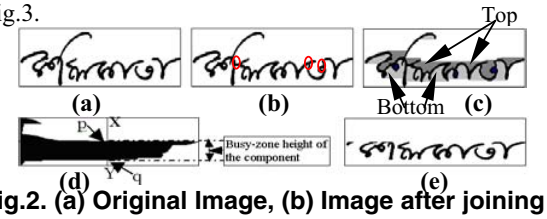

Fig.2. (a) Original Image, (b) Image after joining (c) Image after filling of top, bottom reservoir and loop, (d) Computation of busy-zone, (e) Busy-zone area of the image (a).

2.3.2. Matra/Shirorekha based feature: Before going to matra/Shirorekha feature we shall discuss here about busy-zone. Busy-zone of a word is the region of the word where a maximum portion of its characters lie, and is extracted as follows. First all top and bottom reservoirs are detected and those satisfying certain threshold and loops are filled. Next based on horizontal projection profile of this filled-up image the busy-zone of a word image is calculated as shown in Fig.2(a-e).

The longest horizontal run of black pixels of the busy-zone of a Bangla word will be much longer than that of English script. This is so because the characters in a Bangla word are generally connected by matra/ Shirorekha (see Fig.4 where row-wise histogram of the longest horizontal run is shown in the right part of the words). Matra feature is considered to be present in a word, if the length of the longest horizontal run of the word is (a) greater than 45% of the width of a word, and (b) greater than thrice of the height of busy-zone.

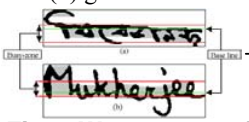

Fig.3. Water reservoir based feature. (a) Bangla word, (b) English word.

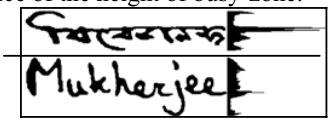

Fig.4. The matra/ Shiroreka feature in Bangla and English word is shown.

2.3.3. Features based on the position of small component: Here we consider all the components whose height and width is less than twice of the R_w (stroke width) and compare their position with respect to the busy-zone. If such components lie completely above or below the busy-zone then those component number and position is used as a feature. This feature is selected because in English we find some characters with disjoint upper part (like dots of i and j) and in Bangla also we find some characters with disjoint lower part (like dots of র্ড় etc.). Based on the above features we use a tree-classifier for word-wise handwritten script identification.

2.4. Pin-code box detection and pin-code numeral extraction

In some Indian postal documents (e.g. Post-card, Inland letters etc.) there are pre-printed boxes (known as pin-code box) to write the pin-code. Here, at first, we detect presence of pin-code box. If it exists, the pin-code and the numerals from the pin-code are extracted as follows.

For pin-code box extraction we apply component labeling and select those components as candidates,

whose length lies between five to seven times the width of the component. We scan each column of a selected component from top and as soon as a black pixel is reached we stop and note the row value of this point as t_i, where i is the column index. Similarly, we scan from bottom to get b_i for the same column. We compute $|b_i - t_i|$, for all columns. Let W be the width of the component (number of column). The selected component satisfying $|(b_i - t_i) - 2R_w| \leq W \leq |(b_i - t_i) + 2R_w|$, for all i=1 to W, is chosen as pin-code box component.

After detection of the pin-code box, vertical and horizontal lines are detected and deleted. Next, depending on the positions of the vertical lines the pin-code numerals are extracted. Pin-code box extracted from Fig.1 is shown in Fig.5(a). Also pin-code numerals extracted from Fig.5(a) are shown in Fig.5(b).

(a) (b)

Fig.5. (a) Extracted part of pin-code box from DAB of Fig.1. (f) Extracted pin-code numerals from the pin-code box.

3. Numeral recognition

3.1. Disconnectivity removal of numeral

Sometimes because of poor document a numeral may be broken. Analyzing the morphological structure features of the numerals, broken parts of the numerals are connected, to improve recognition performance [5].

To generate the structural features, we use contour smoothing and linearization [5]. For any component of the image, using Freeman chain code based contour-tracing algorithm the contour of component is extracted. The contour is then smoothed and converted to line consisting of ordered pixels. Next, depending on the value of the direction codes of two consecutive lines the structural codes are assigned to the start or end points of the linereazied lines of the contours. The structural points describe the convex or concave change in different chain code direction along the contour and are used to represent the morphological structures of a contour. After detection of the structural points, the binary image is thinned to get the end points and junction points. Next, based on some predefined criteria we select a pair of end points for joining and the broken parts of the numeral are joined via a line of width equal to R_w. For illustration, see Fig.6.

After removing the disconnectivity of numerals, we proceed for recognition. We do not compute any feature from the image. The raw images normalized into 28x28 pixel size are used for classification.

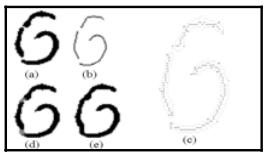

Fig.6. Disconnectivity removal of numeral. (a) the original image, (b) the thinned of the image, (c) the contour with structural points, (d) image after joining (joined part is shown in gray), (e) the final binary image after joining of the broken part.

3.2. Normalization

Normally, in normalization the character image is linearly mapped onto a standard plane by interpolation/ extrapolation. Here we use an Aspect Ratio Adaptive Normalization (ARAN) technique for our purpose [6]. Example of original and normalized image is shown in Fig.7.

3.3. Neural network

We used a Multilayer Perceptron (MLP) based scheme [6] for the recognition of English and Bangla numerals. Because of bi-lingual (English and local language Bangla) nature of the Indian postal documents the number of numeral class is supposed to be 20, but we have used only 16-classes. This is because of shape similarity of English and Bangla 'zero', English 'eight' and Bangla 'four'. Moreover English and Bangla 'two' looks sometimes very similar. English 'nine' and Bangla 'seven' are also similar. To get an idea of such similarity see Fig.8.

In the proposed system we used three classifiers for the recognition. The first classifier deals with 16-class problem for simultaneous recognition of Bangla and English numerals. The other two classifiers are for recognition of Bangla and English numerals, individually. Based on the output of the 16-class classifier we decide the language in which pin-code is written. As Indian pin-code contains six digits, if the majority of these six numerals are recognized as Bangla digits the Bangla classifier is used otherwise the English classifier is used.

4. Recognition of city names

4.1. Word recognition

Our approach of recognizing handwritten city names is based on a Hidden Markov Model (HMM) which is combined with Markov Random Field (MRF). It operates on pixel level in a holistic manner over the whole word which is viewed as outcome of the MRFs.

Compared to other HMM approaches employed for handwriting recognition which are 1D systems, this context based approach is a fully 2D model. We choose such a model because handwriting is essentially two-dimensional in nature.

However, direct extension of HMM into two dimensions leads to NP-hard computational complexity. Among several alternatives suggested for computationally tractable solutions, Planar HMM, Markov Random mesh and the Non-Symmetric Half Plane (NSHP) Markov chains are proposed [3,7].

(a) (b)

Fig.7. Example of (a) Original Images (b)Normalized Images.

(a) (b)

Fig.8. (a) English Nine and Bangla Seven, (b) English and Bangla Two

4.2. Formal description of the NSHP-HMM

The model works on height normalized binary image of the word, which is considered as one possible realization of the Markov random field. For illustration see Fig.9. The NSHP at pixel position (i, j) defined as Σ_{ij} is given by

$$\Sigma_{ij}=\{(k,l)\in L,|l<j \text{ or } (l=k,k<I)\} \quad (1)$$

where *L* is the lattice of pixels defining the word image. Usually the bounding box of normalized image is considered as *L*. The Markov chain is defined over a neighborhood Θ_{ij}. Various type of neighborhoods can be considered for the NSHP. One example of the neighborhood Θ_{ij} of (i, j) is given in Fig 9.

Now, let us define a random field $X=\{X_{ij}\}$ where $(i,j)\in L$. The column *j* of the field is denoted as X^j. Let the grey value at x(i,j) be x_{ij}. We define conditional probability $P(X_{ij}|X_{kl})$ as the probability of realization of x_{ij} at (i,j) given that the grey value at (k,l) is x_{kl}.

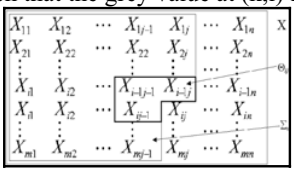

Fig.9.The NSHP Hidden Markov model.

The Markov process is defined to be dependent only on the neighborhood Θ_{ij} i.e.,

$$P(X_{ij}|X_{\Sigma ij})= P(X_{ij}|X_{\Theta ij}) \quad (2)$$

The probability of random field X denoted as P(X) is written as the product from over all column fields, which in turn is written as the product of individual pixel probabilities over the column, i.e.,

$$P(X)=\prod_{j=1}^{n}P(X^j|X^{j-1}\dots X^1)=\prod_{j=1}^{n}\prod_{i=1}^{m}P(X_{ij}|X_{\Sigma ij})=\prod_{j=1}^{n}\prod_{i=1}^{m}P(X_{ij}|X_{\Theta ij}) \quad (3)$$

It is assumed here that the conditional probabilities are independent. Now, a HMM denoted by λ is introduced. So given a model λ, the probability of X is given by

$$P(X|\lambda)=\prod_{j=1}^{n}\prod_{i=1}^{m}P(X_{ij}|X_{\Sigma ij},\lambda) \quad (4)$$

The HMM is defined by the following parameters: $S=\{s_1,\dots,s_N\}$ which are N states of the model, where $q_j \in S$ denotes the state associated with column X^j. Also define the state transition probability matrix $A = \{a_{kl}; 1\le k,l\le m\}$ where a_{kl} is the transition probability from *k* state to *l* state. The initialization is done by the initial state probability $\pi=\{\pi_i;\ 1\le i\le m\}$. Finally, there is a conditional pixel observation probability $B=\{b_{il}\ ;\ 1<i<n;\ 1<l<m\ \}$ where

$$b_{il}(x,x_1,x_2..,x_p)=P(X_{ij}=x|x(\Theta_{ij}),q_j=s_l) \quad (5)$$

i.e., the probability that the current pixel is of value *x* given the neighborhood pixels as well as the state *j* is s_l. Briefly, the model is characterized by

$$\lambda=(\Theta, A, B, \pi) \quad (6)$$

For the training phase the goal is to determine the parameters of A and B as well as π that maximize the product $\Pi_r^k = P(X_r|\lambda)$ where X_r denote a training pattern image. This is done by the well-known Baum-Welch re-estimation procedure. In this way, the model λ for each pattern class in trained sub-optimally.

In the test phase the class, for which the likelihood is maximum, is chosen. In other words X comes from the model λ^* where

$$\lambda^*=\underset{\lambda\in\Lambda}{\operatorname{argmax}}P(\lambda|X)=\underset{\lambda\in\Lambda}{\operatorname{argmax}}\frac{P(X|\lambda)P(\lambda)}{P(X)}=\underset{\lambda\in\Lambda}{\operatorname{argmax}}P(X|\lambda)P(\lambda) \quad (7)$$

Here Λ denotes the set of models for all classes.

5. Results and discussion

5.1. Results on Preprocessing

The accuracy for postal stamp/seal detection and DAB location are 95.98% and 98.55%, respectively. Some errors appeared due to overlapping of postal stamp/seal with the text portion of address part. Some errors also appeared due to poor quality of the images.

For the experiment of script identification we use a database of 2342 (1100 Bangla and 1242 English) handwritten words and 650 (400 Bangla 250 English) printed words collected from postal documents. From the experiment we notice that our proposed scheme has an accuracy of about 89% on handwritten data and 98.42% on printed one.

The performance of the proposed system on pin-code box extraction on 2860 postal images is 97.64%. The main source of errors was due to broken pin-code box,

poor quality of the images and touching of the text portion of DAB with the pin-code box.

5.2. Result on numeral recognition

We collected 15096 (7869 of Bangla and 7227 of English) numerals for experiment of which 80% data were collected from postal documents. Among these numerals 8690 were selected for training of the proposed 16-class recognition system and the remaining were used as test set. These data were also used for experiment on individual classifiers. The overall accuracy of the proposed 16-class classifier and individual Bangla and English digit classifiers on the above data set are given in Table 1. Although the result of our classifier on English numerals of Indian pin-code is only 93.0%, which is not attractive, we test this system on the MNIST data set to get a comparative result. And we obtained 98.6% accuracy on English digit classifier.

From the experiment we noted that the most confusing numeral pair was Bangla 'one' and Bangla 'nine' (shown in Fig.11 (a)). They confuse about 6.3% cases. Their similar shapes rank the confusion rate at the top position. Second confusion pair is Bangla seven and English seven (see Fig.11 (b)) with confusing rate 5.3%.

Table 1: Overall numeral recognition accuracy.

Classifier	Recognition rate for	
	Training Set	Test Set
16-class classifier	98.31%	92.10%
English classifier	98.50%	93.00%
Bangla classifier	98.71%	94.13%

(a) (b)

Fig.11. Examples of some confused handwritten numeral pairs. (a) Bangla one and nine (b) Bangla seven and English seven.

5.3. Results on word recognition

For the experiment of word recognition result, we consider Indian city names written in Bangla script. The normalization of the words is performed just in height since left-right NSHP-HMM model can take care of the width normalization, as reported in [9]. Result of differential height normalization is shown in Fig.12.

The overall recognition accuracies for different vocabularies are given in Table 2. 92.04% accuracy is obtained when we consider just 30 classes. Recognition rate decreases if number of word class increases. 86.44% recognition rate is obtained from 76 word-class.

From the experiment we note that the main confusions occur in cases where the word shape is almost similar or in cases where a considerable part of the shape is similar. The other confusion types can be explained with the great variability of the letters and inter-letter connections. While for Latin scripts 52 different letters can be considered, in Bangla there are about 350.

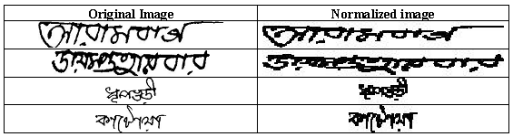

Fig.12. Result of height normalization.

Table 2: Overall recognition accuracy on the training and test set of Bangla data for different vocabulary size (word class).

Word class	Recognition rate		Word class	Recognition rate	
	Training	Test		Training	Test
30	93.49%	92.04%	40	93.41%	90.38%
50	94.00%	88.97%	60	94.02%	88.27%
70	94.58%	87.30%	76	94.83%	86.44%

6. References

[1] R. Plamondon and S. N. Srihari, "On-line and Off-line Handwritten Recognition: A comprehensive Survey", IEEE Trans. on PAMI, Vol. 22, pp. 62-84, 2000.

[2] S. N. Srihari, and E .J. Keubert, "Integration of Hand-Written Address Interpretation Technology into the United States Postal Service Remote Computer Reader System", In Proc. of Forth ICDAR, pp. 892-896. 1997.

[3] Kornai, "An Experimental HMM-Based Postal OCR System", Proceedings of ICASSP'97, pp. 3177-3180, 1997.

[4] K. Roy, A. Banerjee, and U. Pal, "A System for Word-wise Handwritten Script Identification for Indian Postal Automation", IEEE INDICON-04, pp. 266-271, 2004.

[5] Donggang Yu and Hong Yan, "An efficient algorithm for smoothing linearization and detection of structural feature points of binary image contours", PR Vol. 30, pp. 57-69, 1997.

[6] K. Roy, S. Vajda, U. Pal, and B. B. Chaudhuri, "A System towards Indian Postal Automation", Proc. of 9th IWFHR, pp. 361-367, 2004.

[7] O. E. Agazzi and S. Kuo, "Hidden Markov model based Optical Character Recognition in the presence of Deterministic Transformation", PR, Vol. 26, pp. 1813-1826, 1993.

[8] G. Saon and A. Belaïd, "High Performance Unconstrained Word Recognition System Combining HMMs and Markov Random Fields", IJPRAI, Vol. 11, pp. 771-788, 1997.

[9] C. Choisy and A. Belaïd, "Handwriting Recognition using Local Methods for Normalization and Global Methods for Recognition", In Proc. of 6th ICDAR, pp. 23-27, 2001.